Fusion of Multiple Cues from Color and Depth Domains using Occlusion Aware Bayesian Tracker

Kourosh MESHGI † Shin-ichi MAEDA † Shigeyuki OBA † and Shin ISHII †

† Graduate School of Informatics, Kyoto University, Gokasho, Uji-shi, Kyoto, 611-0011 Japan

E-mail: † {meshgi-k,ichi,oba,ishii}@sys.i.kyoto-u.ac.jp

Abstract Object tracking has attracted considerable attention recently because of high demands for its everyday-life applications. Appearance-based trackers had a significant improvement during the last decade, however they are still struggling with some challenges that are not addressed completely so far. Tackling background clutter, abrupt changes in target movement, sudden illumination changes and varying scale of the target are the main design goal for many approaches, while occlusion are often left aside due to its complexity. We proposed an occlusion aware Bayesian framework which deals with occlusion in a way that search area for occluded object expands rapidly, that grants trajectory independence and quick occlusion recovery to the algorithm. Furthermore the algorithm employs multiple cues from color and depth domains to have a robust result against illumination changes and clutter. The Bayesian framework is modified in a way to accommodate this probabilistic fusion. Applied to Princeton RGBD Tracking dataset, the performance of our method is discussed and compared with the state-of-the-art trackers.

Keyword occlusion handling, RGBD particle filter, multiple cues

1 Introduction

Object tracking is taking a new turn with the advent of cheap RGBD sensors, in which the depth data is obtained using active sensing using methods such as time-of-flight range detection or structured lighting scanning. This new piece of information, augment 2D images and indicates a new pathway to enhance the understanding of man-made systems from the world around them. The use of depth maps has shown improvement in object segmentation, recognition and tracking, however, human visual system still outperform machines especially in dynamic scenarios with obstructions or illumination changes. So perhaps computer vision has just not attempted the same level of sensor fusion that is necessary for scene comprehension similar to human capability. Cameras operate at 100 megapixels today (similar to rod and cone count in the human eye) and the computation power is advancing each day, so the advancement could be achieved in more sophisticated intelligent vision inspired from human visual system. Psychologists believe that the astonishing performance of human visual system is based on agglomerating different visual cues. For instance [3] enumerates and discusses the nine cues for depth perception. Also [4] proposed a weak fusion of different cues (stereo vision disparity, shading, texture, motion parallax, kinetic depth effect and occlusion) as the underlying mechanism of robust depth perception in human. These findings suggest that using multiple cues, any vision mechanism could disambiguate depth-related problem including occlusions and improve the accuracy of tasks like depth measuring and object tracking.

Specifically for object tracking several attempts has been made to fuse multiple cues using the appearance models. In previous works on RGB, the color, texture, motion and shape cues has been used to achieve a higher tracking accuracy than each of them in isolation [17]. Furthermore, a few methods have been proposed to track targets by learning object specific appearance model using different cues given an initial position. Such methods are sensitive to the failure of their detectors and are subjected to model drift.

In order to improve robustness and accuracy, multiple approaches explore the idea of injecting knowledge about 3D structure of the scene into the tracking process using depth maps. For example [18] proposed a pedestrian tracking algorithm using a combination of histogram of gradients on color image and depth map with multiple Kinect cameras. Also in [16] a skeleton tracking method based on Random Forest is proposed which uses RGBD data. Another paper by [13] copes with the challenges of object tracking, by integrating multiple complementary sensing modalities and perceptual cues acquired using an ensemble of detectors in color and depth domains. Image-based detectors ranges from pedestrian detector to upper body, face, and skin detector, and depth-based detectors includes shape detector and motion detector. Using a sampling based method constructed on a tracking-by-detection formulation, these detectors are integrated to obtain a final tracking result. In another RGBD paper [6] the shape parameters in 3D, the ratio of points in a voxel, and the motion was used. In this paper an explicit occlusion detection method is also proposed based on the

premise that in the occlusion state, the depth histogram inside bounding box is expected to have a newly rising peak with a smaller depth value than target, and/or a reduction in the size of bins around the target depth. This technique then use a classifier to obtain the final tracking result which makes it prone to model drift in rapidly changing environments.

In this paper we propose an extension to occlusion-aware particle filter proposed in [2] to enhance it with multiple complimentary cues in the image and depth domains. This framework is robust against model drift and handles occlusion and by addition of several cues, the system promises to achieve high performance. Albeit using computational complex features such as HOG negates the real-time property of this framework, yet it empowers the system with accurate cue about the presence of target in one of the hypotheses.

2 Proposed Method

The model we propose here is an extension to the framework proposed in [2]. This framework, occlusion-aware particle filter, by exposing the dynamics of particle filter to a stochastic binary occlusion flag, handles persistent occlusions, grants a quick recovery to the system, and enable fusion between color and depth channels. Initially based on a two features, histogram of colors and median of depth, the method showed a good performance, however, in some scenarios these features turn out to be inefficient or inadequate. Thus other features should be considered to mitigate these shortcoming to improve tracking accuracy and bring additional robustness to tracking. In this regard, the model is extended to enable multiple feature data fusion. Assuming independence between extracted features, the idea is to combine them in log-likelihood fashion to achieve minimal yet stabilized computation. The features are collected from different domains which can explain a single scene: colors, textures, edges, 3D structure, and depth.

2.1 Overview of the Proposed Model

A particle in the occlusion-aware particle filter framework [2] is composed of a bounding box coordinates B_t and a flag Z_t which indicates the occlusion condition in the box, $X_t = \{B_t, Z_t\}$. An RGBD observation with sensors such as Microsoft Kinect is composed of two channels as well, the 2D color image $I_{t,rgb}$ and depth map $I_{t,d}$, and together shapes a 2.5D observation in time *t* in. We define Y_t as a patch of I_t embodied in bounding box defined by B_t which will have the form $Y_t = g\ (I_t; B_t) = \{Y_{rgb,t}, Y_{d,t}\}$. The target model θ_t which is a feature vector combining all features, is calculated on first frame using the given bounding box, and serves as the basis for matching particles in each frame. The observation model of the particle filter then will be divided to two components, occlusion case $p(Y_t \mid B_t, Z_t = 1, \theta_t)$ and no-occlusion case $p(Y_t \mid B_t, Z_t = 0, \theta_t)$.

$$p(Y_t \mid X_t) = (1 - Z_t)\, p(Y_t \mid B_t, Z_t = 0, \theta_t) + Z_t\, p(Y_t \mid B_t, Z_t = 1, \theta_t) \quad (1)$$

The probability of occlusion case follows a uniform distribution while the no-occlusion case is derived from combining features. This choice enables the particle filter to follow the target by evaluating multiple hypotheses based on features. For this calculation however, only the particles contributes which are not marked for being suspected to occlusion. The marked particles are all assigned a similar likelihood, since they are suspected to be occluded and the features cannot extract meaningful information from them.

$$p(Y_t \mid B_t, Z_t = 1, \theta_t) \propto 1 \quad (2)$$

Following the assumption of feature independence, the no-occlusion case probability is represented as

$$p(Y_t \mid B_t, Z_t = 0, \theta_t) \propto p_1(Y_t \mid B_t, \theta_{1,t}) \cdots p_n(Y_t \mid B_t, \theta_{n,t}) \quad (3)$$

in which *n* is the number of features in a feature set F. All the probabilities are calculated by comparing the feature f_i extracted from the observation patch induced by the bounding box of the particle with the respective section of template feature vector θ_t. Furthermore $\theta_{i,t}$ denotes the section of template which keeps the values for feature *i* at time *t*. Using the similarity measure D_i for feature f_i, no-occlusion case probability is written as

$$p(Y_t \mid B_t, Z_t = 0, \theta_t) \propto \exp\left(-\sum_{i=1}^{n} \sigma_i D_i\left(f_i(Y_t), \theta_{i,t}\right)\right) \quad (4)$$

in which the term σ_i is weighting factor for each feature (which is equal for all features in this implementation). Essentially a particle filter, the framework then estimates the new target position $\hat{B}_t$ and the occlusion flag $\hat{Z}_t$ by calculating expectation over all particles.

$$\hat{B}_t = \mathbb{E}(B_t) \cong \sum_{i=1}^{N} p(B_{i,t}) B_{i,t} = \sum_{i=1}^{N} p(X_{i,t} \mid Y_{i,t}) B_{i,t} \quad (5)$$

$$\hat{Z}_t = \mathbb{E}(Z_t) \cong \sum_{i=1}^{N} p(Z_{i,t}) Z_{i,t} = \sum_{i=1}^{N} p(X_{i,t} \mid Y_{i,t}) Z_{i,t} \quad (6)$$

If the occlusion flag value exceeds a certain threshold δ_{occ}, the next target is marked as occluded. It should be noted that the occlusion flags are assigned stochastically, but using the

resampling mechanism of particle filter which select surviving particles based on their probability, their estimation of whether a particle is likely to be occluded or not is corrected through the course of time. The resampling treat the occlusion flag with a state transition model, as well as the bounding boxes.

$$p(X_{t+1} \mid X_t) = p(B_{t+1}, Z_{t+1} \mid B_t, Z_t) = p(B_{t+1} \mid B_t)\, p(Z_{t+1} \mid Z_t) \quad (7)$$

Finally the template is updated (separately for each feature *i*) using a leaky memory scheme parameterized by forgetting factor λ_i, unless an occlusion state detected. In this equation $\hat{Y}_t$ is the patch of frame embodied by the estimated bounding box $\hat{B}_t$.

$$\theta_{i,t+1} = \begin{cases} \theta_{i,t} & , \hat{Z}_t > \delta_{occ} \\ \lambda_i f_i(\hat{Y}_t) + (1-\lambda_i)\theta_{i,t} & , \hat{Z}_t \le \delta_{occ} \end{cases} \quad (8)$$

It should be noted that in order to block the effect of motion model on the results of particle filter, a random walk motion model is employed in this paper.

2.2 Features

In this manuscript we used several features, to boost the performance and improve the robustness and resilience of the system. Each feature has a similarity measure working best for it to encourage similarity with the template while discouraging similarities with other parts of the scene. We tried to use best practices for most of them.

COLOR: Being both compact and invariant to specific changes in the image, color histograms gained popularity in object tracking, e.g. the famous mean-shift tracker [7] and color-based particle filter [1] is built upon this feature. Another trend to use color histograms is to model them as Gaussian mixture models in combination with multi-hypothesis adaptive models [8]. In this implementation we use normalized color histograms to exploit color data and following the result of [5] the choice of (dis)similarity measure would be KL-divergence.

TEXTURE: There are plenty of methods used to describe texture of a region in the scene. Histograms moments, co-occurrence matrices, Fourier derived methods and auto-correlation, histogram of local binary patterns [9] and wavelets especially Gabor functions have been largely used in the literature. In this implementation we used histogram of quantized local binary patterns with chi square (χ^2) distance as the similarity measure.

EDGE: Comparing images using only edge information as an orientation and scale invariant template-matching task has been addressed in many studies including [10] which uses Hausdorff distance.

3D SHAPE: 3D point cloud as a result of pixel-wise integration of color and depth channels, encompassed a wealth of information which can be exploited to bring higher accuracy to the tracking system. Gridding the cloud into regular voxels, each cell contains several points of target. These points are then used to calculate several 3D shape features [11] including scatter-ness, linear-ness and surfaceness (inspired by Spin Images). Also the number of points in each voxel is concatenated to this feature vector. We use L2 norm to compare these scalars.

DEPTH: Being augmented with depth, the scene can be expressed in more details using depth information along with appearance based features. Depth features have been proved to be efficient in compensating failures of appearance feature, but the depth values themselves are expressive in many scenarios [15, 18]. To harvest such information, a histogram of depths with coarse binning is used to help implicit object segmentation [12], as well as exploiting the temporal depth consistency in the sequence to keep the focus on the target. The depth histograms are compared using the Bhattacharyya distance because of its nice statistical properties.

3 Evaluations

To evaluate our algorithm we use a sequences captured using Kinect, the RGBD sensor from Princeton Tracking database [6]. The scenario consist of a moving object with several self-occlusions and a persistent dynamic full occlusion. The video also suffers from background clutter and scale changes, while in-plane rotation and non-rigid deformation challenges trackers. Additionally, the texture is sometimes distorted with motion blur and slight shadow cast and shading are also present in the sequence. There is another dynamic occluding object with same colors of the subject and the depth patterns of two subjects swap slightly during the sequence.

To see the effect of each of the features, several versions of our code is compared together incrementally:

A. particle filter tracker using edge cues;
B. adding histogram of color (HoC) to (A);
C. adding histogram of depth (HoD) to (B);
D. adding texture cues (LBF)to (C);
E. adding 3D shape features to (D);
F. and finally, using all the features in the occlusion aware particle filter (OA PF).

Comparing results are done using area under curve of success plots, scale adaptation error and central point error. Following the style of [6] we define the success *S* as

$$S = \begin{cases} \left|B^*_{t+1} \cap \hat{B}_{t+1}\right| \Big/ \left|B^*_{t+1} \cup \hat{B}_{t+1}\right| & , \hat{Z}_{t+1} = Z^*_{t+1} = 0 \\ 1 & , \hat{Z}_{t+1} = Z^*_{t+1} = 1 \\ -1 & , \hat{Z}_{t+1} \neq Z^*_{t+1} \end{cases} \quad (10)$$

where $|.|$ denotes the number of pixels in the defined region, the tracked bounding box is denoted by $\hat{B}_{t+1}$ and the ground truth bounding box B^*_{t+1}. To measure the tracker success on a sequence of frames, we count the number of successful frames whose overlap S is larger than a given overlap threshold t_o. Using one success rate value at a specific threshold (e.g. t_o=0.5) for tracker evaluation may not be fair or representative and the area under curve (AUC) of success plot will be used in the evaluation table. Central point error and scale error are defined as the L2-norm difference of *(x,y)* position and *(width,height)* pair values of the estimated bounding box and ground truth respectively. Having normalized them, the average values of these metrics are presented in Table 1, the evaluation table, which gathers the algorithm performances together. Figure 1 illustrates different aspects of the trackers for better comparison.

As Figure 2 depicts, tracker (A) although performs well initially, cannot continue tracking very soon. The reason is that the subject moves toward an area of the scene in which the background edges are prominent and the edge clutter induce noise to many of the particles impairing the edge-based tracker. Tracker (B) by the help of color cues keeps the track on the subject, but after disappearance of subject behind the occluding pedestrian which has similar colors, it is drifted toward the occluding pattern. This is known weakness of color histograms to handle same-color objects and in this scenario, the edges could not solve the problem. Tracker (C) however, introduces a powerful feature from depth domain to compensate for background clutter (both color and edge) and same-color objects. Although there exist depth clutter in the scenario as well (Figure 3), still histogram of depth handles it using fine-binning. After the occlusion, the tracker suffers from two other features willing to track the occluder, and didn't recover quickly.

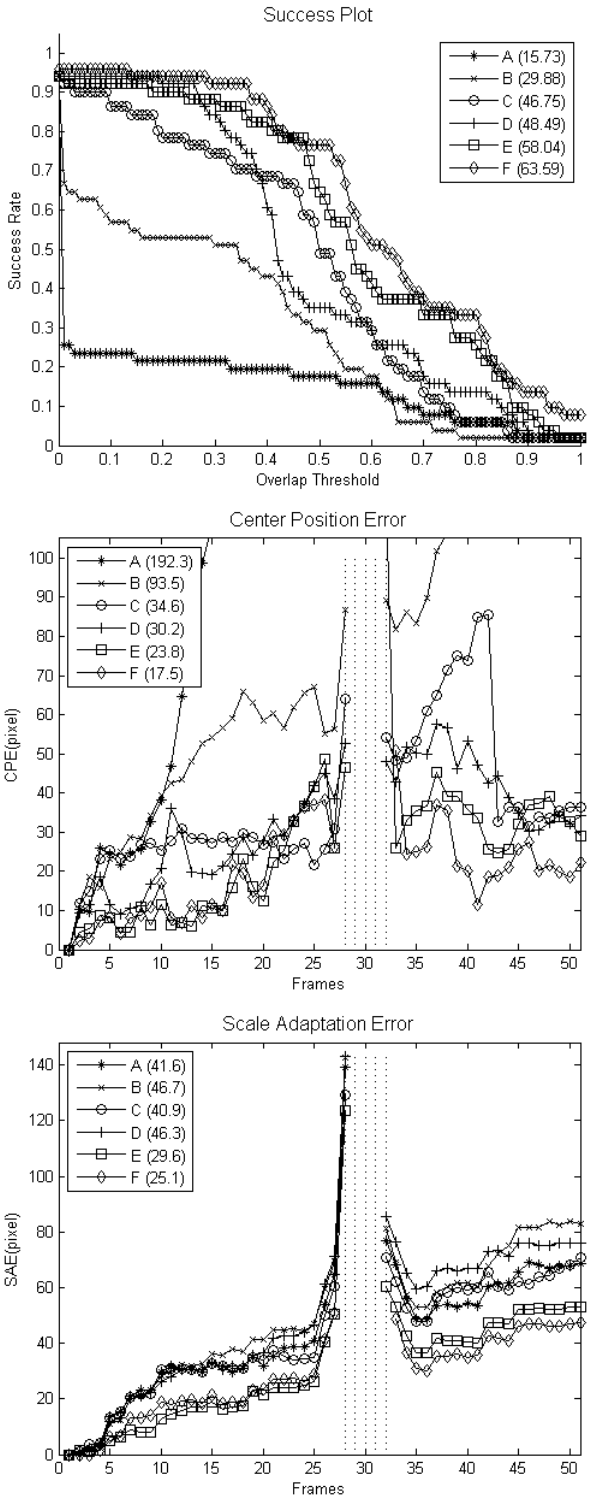

Fig 1. Success Plot, CPE plot, SAE plot (dotted area indicates occlusion)

Fig 2. Qualitative Comparison. Color codes: white dashed=A, black dashed=B, white dash-dotted=C, black dash-dotted=D, white solid=E, black solid=F, dark gray solid=Ground truth. **(From Left To Right, Top to Bottom)** Frame1: All trackers are initialized in the same position, models initiated. –Frame16: Tracker (A) was trapped in background edge clutter. –Frame28: The target is partially occluded, tracker (F) decided to announce the occlusion state to prevent model drift. –Frame30: Full occlusion: all trackers but (F) are updating their models with wrong template i.e. the occluder. –Frame32: End of full occlusion, tracker (E) still searches the same position that loses the target relative to occlude position, tracker (F) still maintains the occlusion state. –Frame34: Recovery from occlusion, tracker (F) with intact models recovers quickly. –Frame40: Tracker (B) lost the main target and start to track occluder due to model drift. –Frame42: Some trackers such as tracker (C) are experiencing fluctuations due to model drifts, yet strong cues such as histogram of depth help them to continue tracking. –Frame50: Trackers lock on the subject again, but tracker (F) does not suffer from model drift and has better tracking performance. (The sequence image is washed out for image clearance. The results are available in color in the first author's webpage).

Tracker (D) by introducing the texture clue, improves the scale adaptation of the system since it rejects the particles having part of background in their scope e.g. having structure ceiling in the bounding box. Yet, the feature is drifted after the occlusion toward the occlude subject and weaken the effect of histogram of depth even more. This results in sudden fall in the success plot. Larger slope of a curve in this plot indicates larger localization error, meaning that tracker is able to follow the target but with low precision on bounding box location or size. The scale adaptation plot in this case, shows improvement in scale adaptation after occlusion and backs our claim. However, due to stochastic nature of occlusion flag, the scale adaptability and localization of Tracker (F) is sometimes better than (E) and sometimes worse, but in the average case, it works on average as good as tracker € before occlusion. Tracker (E) with using the computational complex shape features, improves the precision and scale adaptation of the tracker to a great extent. Finally using the stochastic occlusion handling in tracker (F), the tracker recovers from the occlusion more quickly benefiting from growing search area. Additionally, it handles model drift so that the scale adaptation and localization ability of the algorithm improves significantly.

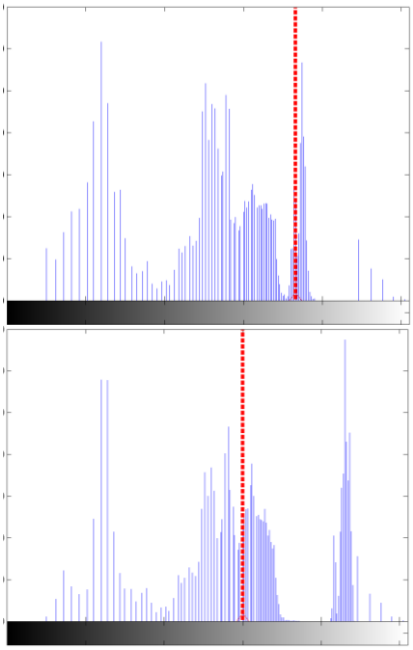

Fig 3. Depth Clutter in the scenario (darker = closer to camera). The target is marked in histogram of depth of frame 1(top) and frame 51(bottom). Targets swap their depth values in the middle of the scenario. That emphasizes the role of model update in the scenario as well as the importance of preventing model drift.

Based on the qualitative observations and plots in Figure 1 it can be inferred that introducing new features to the framework, if chosen well, lower the error of the system. Furthermore, the advantage of occlusion-aware particle filter can be seen in quick recovery from occlusion, better scale adaptation (since the irrelevant particles are marked as occluded stochastically), and has tighter grip on the target.

Table 1: Evaluation table, the area under curve of success plot, average central point error, and average scale adaptability error of trackers are compared in this table.

Tracker	AUC	$\overline{\text{CPE}}$	$\overline{\text{SAE}}$
A (edg)	15.7255	192.2745	41.6292
B (edg+hoc)	29.8824	93.4759	46.7128
C (edg+hoc+hod)	46.7451	34.6243	40.8838
D (edg+hoc+hod+tex)	48.4902	30.1808	46.2762
E (edg+hoc+hod+tex+shp)	58.0392	23.8482	29.6286
F (all + occlusion handling)	63.5882	17.4617	25.0734

4 Conclusion & Future Works

In this paper we extend our previous framework, occlusion-aware particle filter tracker, to support arbitrary number of features. Using features from color and depth domain, the algorithm was capable of tracking more accurately than each feature in isolation or a subset of them. The merits of the framework can be enumerated as facilitating the fusion of features, preventing model drift, and quick recovery from occlusion.

The next step toward having a robust and accurate tracker would be: (i) to incorporate more resilient features, (ii) to measure the confidence of each data channel, and (iii) to update of the model adaptively based on information contained in each observation,

References

[1] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An Adaptive Color-based Particle Filter" in JIVC Vol. 21, No.1, pp. 99-110, 2003.

[2] K. Meshgi, Y. Li, S. Oba, S. Maeda, S. Ishii, "Enhancing Probabilistic Appearance-Based Object Tracking with Depth Information: Object Tracking under Occlusion," in IEICE Tech. Rep., vol. 113, no. 197, IBISML'2013, pp. 85-91, Sept. 2013.

[3] J.E. Cutting, and P.M. Vishton, "Perceiving layout and knowing distances: The integration, relative potency, and contextual use of different information about depth," Academic Pr, Vol. 46, pp. 69-117, 1995.

[4] M.S. Landy, L.T. Maloney, E.B. Johnston, and M.J. Young, "Measurement and modeling of depth cue combination: In defense of weak fusion," J. of Vision Research, Vol. 35, pp. 389-412, 1995.

[5] K. Meshgi, and S. Ishii "Expanding Histogram of Colors with Gridding", in IEEE SSP'2014, *in press*.

[6] S. Song, and J. Xiao, "Tracking Revisited using RGBD Camera: Unified Benchmark and Baselines," in ICCV'2013.

[7] D. Comaniciu, V. Ramesh and P. Meer, "Real-Time Tracking of Non-Rigid Objects using Mean Shift," in CVPR'2000, pp. 142-149.

[8] M. Isard and J. MacCormick, "Bramble: A Bayesian Multiple-Blob Tracker," in ICCV'2001, pp. 34–41.

[9] T. Ojala, M. Pietikäinen, and M. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," in PAMI, 2002.

[10] D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge, "Comparing images using the Hausdorff distance," PAMI, Vol. 15, no. 9, pp. 850-863, 1993.

[11] A. Johnson. "Spin-Images: A Representation for 3-D Surface Matching," PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997.

[12] E. Parvizi, Q.M.J. Wu, "Multiple Object Tracking Based on Adaptive Depth Segmentation," in CRV'2008, pp. 273-277.

[13] W. Choi, C. Pantofaru, and S. Savarese, "Detecting and tracking people using an rgb-d camera via multiple detector fusion," in ICCV'2011, pp. 1076-1083.

[14] C. Wojek, S. Walk, S. Roth, and B. Schiele, "Monocular 3D scene understanding with explicit occlusion reasoning," in CVPR'2011, pp. 1993-2000.

[15] M. Luber, L. Spinello, and K. Arras, "Learning to detect and track people in rgbd data" in RGB-D Workshop, RSS'2011.

[16] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from a single depth image," in CVPR'2011.

[17] L. Mihaylova, P. Brasnett, N. Canagarajah, and D. Bull, "Object tracking by particle filtering techniques in video sequences," in Advances and Challenges in Multisensor Data and Information, pages (260-268), 2007.

[18] L. Spinello and K.O. Arras. "People Detection in RGBD Data." in IROS'2011, pp. 3838-3843.